

## **REMARKS/ARGUMENTS**

Claims 1-16 are pending of which claims 8-16 have been withdrawn from consideration. In light of the amendments and following remarks, Applicant believes claims 1-7 are allowable.

### **The § 103(a) Rejection of Claims 1-7**

Claims 1-7 were rejected under 35 USC § 103(a) as being unpatentable over U.S. Patent No. 5,832,431, issued November 3, 1998 to Sheffield in view of U.S. Patent No. 5,907,847, issued May 25, 1999 to Goldberg and further in view of U.S. Patent No. 6,108,004, issued August 22, 2000 to Medl. Accordingly, it is asserted that if these three references are combined, all the features of the claims are disclosed. For the following reasons, Applicant respectfully traverses the rejection.

The Office Action states that the claims do not specify that there is no need for an invoking or initiating mechanism for code generation (page 3). Additionally, the Office Action stated that the claims could be vague as to how the code is generated.

In a sincere effort to expedite prosecution, Applicant has amended claim 1 to recite that the executable code is automatically generated from the specified data elements. Support for this amendment can be found, for example, on page 34, lines 7-30. With this amendment, Applicant has made it clear that the executable code for extracting the specified data elements is automatically generated from the specified data elements.

The Office Action acknowledges that Sheffield does not disclose generating executable code from the specified data elements as recited in claim 1. However, the Office Action alleges that Goldberg remedies the deficiencies of the primary reference. Initially, the Office Action states that Goldberg describes that an object can have methods for data input and output, calling this data extraction and storage (page 4).

An object can have a method for data input and output. However, claim 1 recites that executable code is automatically generated from the specified data elements for extracting the data elements. Accordingly, claim 1 does not recite calling an existing method of an object for input or output so Goldberg does not disclose the claimed feature.

The Office Action further states "these methods may be used to generate code for all relevant operations on the object's state" (page 7; citing Fig. 5A, step 504 and col. 8, lines 36-39). A closer review of these sections of Goldberg reveals that step 504 is where an executable form of an object behavior is created. Goldberg describes that the underlying source code can be written in a programming language such as Java. However, it is the programmer that is generating the source code, not a computer implemented method that automatically generates executable code as claimed. It is believed that the Examiner felt that it was unclear that the executable code was automatically generated so the amendment makes it clear that Goldberg does not disclose the features claimed.

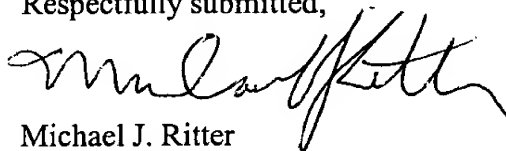
The Medl reference is cited as disclosing data mining and storing data in a database. As such, even if this is the case, the reference does remedy the deficiencies of the other two references as described above.

As the three references, even if combined, do not disclose all the features of claim 1, the references do not support a prima facie case of obviousness and the claim is patentably distinct. The other claims 2-7 are dependent claims so they are patentably distinct for at least the same reasons.

**Conclusion**

For the foregoing reasons, Applicant believes all the pending claims are in condition for allowance and should be passed to issue. If the Examiner feels that a telephone conference would in any way expedite the prosecution of the application, please do not hesitate to call the undersigned at (408) 446-8693.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Michael J. Ritter", written in a cursive style.

Michael J. Ritter  
Reg. No. 36,653

RITTER, LANG & KAPLAN LLP  
12930 Saratoga Ave., Suite D1  
Saratoga, CA 95070  
Tel: 408-446-8690  
Fax: 408-446-8691